Gatekeeper API Documentation

Gatekeeper h4.0

Updated 16/09/2016

Technical Support

support@gymmastersoftware.com

USA: 415 678 1270

Australia: 03 9111 0323

New Zealand: 03 974 9169

The Default enpoints are located at:

$CLIENT_NAME.gymmasteronline.com/gatekeeper_api/v1/$ENDPOINT_NAME/

Where:

$CLIENT_NAME is the name of the client

$ENDPOINT_NAME is the name of the endpoint as specified below

Receiving Data from the Gatekeeper APA

System:

This shows database configuration type data that is system wide (e.g. site independent)
This includes open hours of gyms, warning limits, GymMaster version number, etc.

Fields and Subfields:

- multiclub_hostlist:
  The website host for the other clubs, Note: Seperated by comma.
- roster:
  Defines the roster defining the hours of the gym
- dow:
  Day of Week, Note: 0 is Sunday and 6 is Saturday
- starttime:
  Roster Start Time
- endtime:
  Roster End Time
- name:
  Name of the Roster
- rosterid:
  ID of the Roster
- siteid:
  ID of the site
- schemaversion:
  The GymMaster Database version
- stop:
  Stop on owe amount, (e.g. if member owes over then stop them from coming in)
- stop_incomplete:
  Whether to stop members with incomplete memberships
- warn:
  Warn on owe amount, (e.g. if member owes over then warn them at gate)

Time:

This shows the database current time configuration, an example of this is the database timezone as well as the current time.

Fields And Subfields:

- current_date:
  Current Date of the database
- current_time:
  Current Time of the database
- dow:
  Current Day of week
- epoch:
  Time since epoch
- ts:
  Current Time Stamp

Member:

This shows all the member that have got tags as well as the memberships that the members have, this endpoint shows member and membership specific data, (etc member owing amount, membership startdate, membership enddate, visit count, tagserial).

Fields And Subfield:

- card:
  List the cards that the member has (Note: is the same as tag)
- cardid:
  ID of the cards (Note: is the same as tagid)
- cardserial:
  Serial of the cards (Note: is the same as tagserial)
- gender:
  Gender of the member
- memberid:
  The ID of the member
- membership:
  Lists the membership that the member has
- accountid:
  The membership account
- balance:
  The balance of the membership account

- concession:
  The amount of visits that the member has made
- startdate:
  The startdate of the membership
- enddate:
  The enddate of the membership
- expired:
  Whether the membership is expired
- incomplete:
  Whether the membership has been completed
- membershipid:
  The ID of the membership
- membershiptypeid:
  The ID of the membership type
- refreshdate:
  The day that the balance of the account get refreshed
- tagid:
  The ID of the tag
- tagserial:
  The serial of the tag
- tagserial_short:
  The short version of the tagserial
- name:
  The name of the member
- owe:
  How much the member owes
- owingdeadline:
  The date that the member has to pay the owing amount by (e.g. is allowed in, even if owing is above the stop threshold, up to the date specified)
- siteid:
  The site that the member belongs to
- stopatgate:
  Whether a stop at gate task is active

Membershiptype:

This shows all the programmes that the member can have, membershiptypes/programmes provide all the details and limitations for access to doors.

Fields and Subfields:

- basis:
  The membership type basis name
- basisid:
  The membership type basis ID
- benefitenabled:
  Whether the membership type is using the benefit system
- concessionlimit:
  Specifies a limit on how many concessions are used
- door:
  Specifies a list of doors that this membership type has access to:
- doorid:
  Specifies the doorid of the member
- id:
  The benefit ID
- programmeid:
  The ID of the membership type (NOTE: Programme is Membership Type)
- rosterid:
  The roster ID that gives the access times for the door
- membershiptypeid:
  The ID of the membership type
- multisite:
  Specifies if this membershiptype has multisite
- name:
  Name of the membership type
- priority:
  The priority of the membership
- siteid:
  The ID of the site where this membershiptype is local to

Door:

This provides all the information about the doors.

Fields and Subfields:

- booking_checkin:
  Whether the person should be checked automatically in to the booking
- checkout:
  Whether the door is a checkout only door

- companyid:
  The companyid of the door (Note: Companyid is Siteid)
- concessionhandling:
  Whether the door counts concessions when used
- door_sublocation:
  NOT_USED
- exit_door_id:
  The door ID of the exit door
- gatekeeperid:
  The ID of the gatekeeper that controls the door (Note: Is door controller)
- id:
  Is the ID of the door
- lastupdate:
  The last update timestamp performed on the door
- mac_address:
  The mac address of the gatekeeper
- name:
  The door name
- optional_args:
  The optional arguments of the door (Note: is user configurable)
- reader_type:
  The type of reader that is used
- resourceid:
  The ID of the resource that door is for
- restricted_door:
  If the door is for restricted access
- sentry_com_port:
  The physical path to the reader
- sentry_ip_address:
  The ip address of the reader (Note only applies to network readers)
- sentry_mac_address:
  The mac address of the reader (Note only applies to network readers)
- showlastvisits:
  Shows the visits
- siteid:
  The ID of the site the door belongs to (Note: Siteid is Companyid)
- status:
  The status of the door
- update_user_name:
  The last person who updated the door

- womenonly:
    Whether the door is women only

Sending Data from the Gatekeeper API:

Log_swipe:

To log a swipe we use the log_swipe endpoint, each swipe is of the following format:

```
{
    "swipe":
    [
        {
            "membershipid": INTEGER,
            "tagserial_short": TEXT,
            "comment": TEXT,
            "tagid": INTEGER,
            "tagserial": TEXT,
            "when": TIMESTAMP,
            "doorid": INTEGER,
            "access": INTEGER,
            "memberid": INTEGER,
            "debug": BOOLEAN
        }
    ]
}
```

Now to run through the fields:

- membershipid:
    The id of the membership that the members used or tried to use for the swipe
- tagserial_short:
    The tagserial of the tag using 6 bytes of the tag
- comment:
    The comment that gets used for the door message
- tagid:
    The id of the tag in the database

- tagserial:
    The full tagserial
- when:
    The time of swipe
- access:
    The access code that was given when the tag was swiped (i.e. the swipe result)
- memberid:
    The id of the member
- debug:
    If the swipe was for debugging purposes

The Access field is made up of the following fields:

- granted=1
- doordenied=12
- notimeaccess=13
- offpeakonly=14
- expired=15
- depleted=16
- notstarted=17
- womenonly=18
- stopatgate=19
- concessiononly=20
- incomplete=21
- newcard=23
- overdue=24
- onhold=25
- error=26

Example logged swipes are:

- New Card:


```
{
    "swipe":
    [
        {
            "membershipid": null,
            "tagserial_short": "Mxfe8096",
            "comment": null,
            "tagid": null,
            "tagserial": "Mix0500fe809629c4",
            "when": "2016-09-13 17:42:59",
            "doorid": 5,
            "access": 23,
            "memberid": null,
            "debug": false
        }
    ]
}
```


- Granted:


```
{
    "swipe":
    [
        {
            "membershipid": 55018,
            "tagserial_short": "Mx9e428c",
            "comment": null,
            "tagid": 2,
            "tagserial": "Mix03009e428c29ac",
            "when": "2016-09-13 18:04:38",
            "doorid": 5,
            "access": 1,
            "memberid": 5359,
            "debug": false
        }
    ]
```

```
}
```


- Owing:


```
{
	"swipe":
	[
		{
			"tagserial_short": "Mxaa2d22",
			"memberid": 9112,
			"doorid": 116,
			"membershipid": 54387,
			"access": 24,
			"tagid": 114,
			"comment": "Account is overdue, payment required",
			"debug": false,
			"tagserial": "Mxaa2d22",
			"when": "2016-09-14 11:40:11"
		}
	]
}
```


NOTE: Please be aware as there are foriegn key checks that may reject a visitor entry into the log (e.g. if the membershipid does not belong to a member)